Faster SCALLOP from Non-prime Conductor Suborders in Medium Sized Quadratic Fields

Bill Allombert[1], Jean-François Biasse[2], Jonathan Komada Eriksen[3], Péter Kutas[4,5(✉)], Chris Leonardi[6], Aurel Page[1], Renate Scheidler[7], and Márton Tot Bagi[4]

[1] Inria, Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR5251, Talence, France

[2] University of South Florida, Tampa, USA
[3] KU Leuven, Leuven, Belgium
[4] Eötvös Loránd University, Budapest, Hungary
kutasp@gmail.com
[5] University of Birmingham, Birmingham, UK
[6] ISARA Corporation, Waterloo, Canada
[7] University of Calgary, Calgary, Canada

Abstract. A crucial ingredient for many cryptographic primitives such as key exchange protocols and advanced signature schemes is a commutative group action where the structure of the underlying group can be computed efficiently. SCALLOP provides such a group action, based on oriented supersingular elliptic curves. We present PEARL-SCALLOP, a variant of SCALLOP that changes several parameter and design choices, thereby improving on both efficiency and security and enabling feasible parameter generation for larger security levels. Within the SCALLOP framework, our parameters are essentially optimal; the orientation is provided by a 2^e-isogeny, where 2^e is roughly equal to the discriminant of the acting class group.

As an important subroutine we present a practical algorithm for generating oriented supersingular elliptic curves. To demonstrate our improvements, we provide a proof-of-concept implementation which instantiates PEARL-SCALLOP at record-sized security levels.For the previous largest parameter set, equivalent to CSIDH-1024, our timings are more than an order of magnitude faster than any other SCALLOP version.

1 Introduction

Isogeny-based cryptography dates back to Couveignes' seminal work [27] where he introduced the concept of hard homogeneous spaces, which are today often referred to as cryptographic group actions [1], as a quantum-resistant alternative to the usual Diffie-Hellman key exchange [34]. Cryptographic group actions are a useful tool for designing cryptographic primitives reminiscent of discrete

T. Jager and J. Pan (Eds.): PKC 2025, LNCS 15676, pp. 333–363, 2025.
https://doi.org/10.1007/978-3-031-91826-1_11

logarithm-based primitives that are post-quantum secure. Rostovtsev and Stolbunov [52] rediscovered Couveignes' ideas and the resulting scheme is now dubbed the CRS key exchange. The CRS key exchange utilizes the group action of certain class groups of imaginary quadratic orders on the set of ordinary elliptic curves. The security of the scheme relies on the hardness of inverting the group action. Unfortunately, constructions based on ordinary curves are rather slow.

A breakthrough in this direction was CSIDH [20], where the key idea is to replace ordinary elliptic curves by supersingular ones defined over $\mathbb{F}_p$. On this set of curves, there is a natural group action of the class group of $\mathbb{Z}[\sqrt{-p}]$ that can be utilized to build a key exchange.

De Feo and Galbraith constructed a signature scheme combining CSIDH with the Fiat-Shamir scheme with aborts in a technique called SeaSign [30]. The difficulty (and hence inefficiency) of CSIDH-based signatures is that for cryptographically sized parameters it is hard to compute the structure of the class group. For CSIDH-512, Beullens, Kleinjung and Vercauteren computed the structure of the class group using a record-breaking computation which they then applied to build the signature scheme CSi-FiSh [11]. The framework of CSi-FiSh can also be applied to build threshold signatures [32], ring signatures [10], group signatures [9] and many more cryptographic primitives.

Unfortunately, due to [49] and [15], it is unclear whether CSIDH-512 (and thus CSi-FiSh) achieves NIST level I security, so it is important to have instantiations with larger parameters. Even though CSIDH easily generalizes to higher security levels, CSi-FiSh would require class group computations that are out of reach for current algorithms and computational resources. SeaSign does scale for larger parameter sets but is highly impractical.

The notion of an orientation of an elliptic curve by an arbitrary imaginary quadratic order was introduced to cryptography by Colò and Kohel in their OSIDH protocol [26]. Recently De Feo, Fouotsa, Kutas, Leroux, Merz, Panny and Wesolowski proposed SCALLOP [39] which is a cryptographic group action different from CSIDH/CSi-FiSh that builds on the notion of an orientation.

The key idea of SCALLOP is to use a supersingular elliptic curve oriented by a non-maximal order of large prime conductor in a quadratic number field of small class number, such as prime conductor suborders of $\mathbb{Z}[i]$. The class number of this order can be calculated easily using a standard formula relating the two class numbers. Then computing the structure of the class group reduces to computing certain discrete logarithms in said class groups. By carefully generating parameters, this allows for an implementation of signature schemes for security levels comparable to CSIDH-512 and CSIDH-1024 without the need to use particularly large resources (i.e., pre-computations can be carried out on a laptop). However, from the original construction, it is somewhat unclear whether SCALLOP can be instantiated for security levels comparable to CSIDH-2048 and CSIDH-4096, and SCALLOP is significantly slower than CSIDH.

SCALLOP-HD is a variant of SCALLOP that uses higher dimensional tools developed in [29] to provide polynomial-time parameter generation. The reason

is as follows. In SCALLOP, the natural generator of the order has non-smooth degree. In order to evaluate this endomorphism, it needs to be represented in a compact way. In SCALLOP this is done by writing it as a linear combination of 1 and a smooth degree endomorphism which is a non-trivial task in the parameter generation phase. SCALLOP-HD bypasses this obstacle by representing the isogeny using higher dimensional techniques.

1.1 Our Contributions

We make different design choices compared to SCALLOP for security and efficiency purposes. We use a maximal order with a class number that is large but still efficiently computable (i.e., has a discriminant of roughly 256 bits). We also use a conductor defining a non-maximal quadratic order $\mathfrak{O}$ that is not smooth but also not prime; specifically, it is the product of a few, large primes. Choosing such a conductor defeats all the attacks already considered in SCALLOP and hence does not seem to pose a security threat. Furthermore, using a maximal order with a conductor of this form ensures that its class number will not be smooth, so the method is potentially more resistant against hidden shift attacks.

In the original SCALLOP, the conductor f is chosen to be prime and in such a way that $f \pm 1$ is smooth, in order to utilize the Pohlig-Hellmann algorithm for discrete logarithm computations. This makes the class group computation easy, but becomes hard to achieve for larger security levels. By switching to a product of large primes, we can reduce the class group computations to mid-size discrete logarithm computations in finite fields where this computation is efficient in practice.

The main benefit of this construction is that the group action evaluation is significantly faster than in SCALLOP and SCALLOP-HD. The extra flexibility in our parameter generation facilitates a representation of the orientation by an endomorphism whose degree is a power of 2. In this way, compared to SCALLOP-HD, we do not require higher dimensional isogeny representations and do not need to evaluate higher dimensional isogenies for translating the orientation; explicitly, we replace the $(2^e, 2^e)$-isogenies with 2^e-isogenies. Furthermore, we can use odd degree isogenies in the group action evaluation. As a result, we do not encounter the expensive issue of SCALLOP where the norm of the ideal to be evaluated is not coprime to the norm of the endomorphism that represents the orientation.

As a subroutine, we design a more efficient algorithm for generating oriented elliptic curves together with the orientation. In theory, this can be accomplished in polynomial time using the maximal (quaternion) order to elliptic curve algorithm from [36] or its more practical variant [38]. However, for larger parameter sets, generating a supersingular elliptic curve with prescribed endomorphism ring is computationally very expensive.

Putting all of these ingredients together, we propose a new SCALLOP variant, PEARL-SCALLOP (short for *Parameter Extension Applicable in Real-Life SCALLOP*), that we instantiate for the security levels comparable to CSIDH-512, CSIDH-1024 and CSIDH-1536, and demonstrate a significant practical

speed-up, compared to SCALLOP and SCALLOP-HD. When defining security levels, we will always compare to versions of CSIDH (as the quantum bit security of Kuperberg's algorithm instantiated for class groups is debated). An implementation of PEARL-SCALLOP can be found in the repository https://www.github.com/biasse/SCALLOP-params.

1.2 Technical Overview

Here, we give a more detailed analysis of our technical ideas and make a comparison between SCALLOP, SCALLOP-HD and PEARL-SCALLOP.

It is known [48] that one can instantiate class group actions with any orientation. However, there are three important requirements when designing efficient signature schemes and more advanced primitives:

- Security: Disclosing the orientation should not reveal too much information about the endomorphism ring of the curve;
- Efficient representation: The orientation should have an efficient representation that enables the the evaluation of the class group action;
- Efficiently computable class group structure.

CSIDH satisfies the first two criteria, but its class group structure (or even its class number) cannot be efficiently computed for larger security levels, as CSIDH-512 already entailed a record class group computation. The idea of SCALLOP is to use non-maximal orders of large conductor. In SCALLOP and SCALLOP-HD, the maximal quadratic order has small class number (in the proposed parameters it has class number 1). It would be natural to use a non-maximal order of smooth conductor, as in this case, the orientation would have an efficient representation and an oriented curve could be computed with a single smooth-degree isogeny evaluation. Unfortunately such a construction is insecure because of the following. Let ι be the endomorphism of the curve oriented by the maximal quadratic order. Then the orientation by the non-maximal order corresponds to an endomorphism of the form $\tau = \phi \circ \iota \circ \hat{\phi}$. Now we can evaluate τ on any point of powersmooth degree and then recover ϕ using techniques developed in [33].

In order to avoid such an attack, one can use orders of non-smooth conductor. However, problems arise in satisfying the requirement outlined above. In terms of efficiently representing the orientation, in SCALLOP, this is achieved by using a smooth generator for the underlying order. Such a generator always exists, but making this construction practical is challenging. For instance, finding a smooth generator usually takes subexponential time and the smoothness bounds are highly impractical, which significantly affects the runtime of the group action evaluation. Thus, in SCALLOP, one finds the smooth generator first and then tries to find a suitable conductor f. Since we need $f-1$ to be smooth, this places restrictions on the particular structure of f, which effects the biggest efficiency loss of SCALLOP.

SCALLOP-HD resorts to higher dimensional isogeny representation to address this issue. It uses a prime conductor f where $f \pm 1$ is smooth, but

has no restriction on the smoothness of the generator for the order. This construction allows for polynomial-time class group computation and hence scales well for any security level. The minor drawback here is that one needs to use higher dimensional isogenies for evaluating the orientation and the class group has smooth order (which may or may not be a problem for certain applications or improved hidden shift attacks).

The idea of PEARL-SCALLOP is to use a non-maximal quadratic order of larger discriminant whose class group is still efficiently computable. We use conductors that are not prime but are not smooth; they are a product of a few primes, depending on the security level. We revisit the idea of representing orientations by smooth generators. The advantage now comes from the fact that we search simultaneously for a suitable conductor and the maximal order. Specifically, we find positive integers a and d such that $d + a^2$ (the norm of $a + \sqrt{-d}$) is a fixed power of 2 and the coefficient of $\sqrt{-d}$ in a small power of $a + \sqrt{-d}$ is the product of a few primes. This has two major benefits. First, our orientation is represented via an isogeny whose degree is a power of 2. Second, the class group computation reduces to computing the class group of the maximal order and discrete logarithm computations in moderate sized finite fields. This approach is faster than SCALLOP-HD but does not scale in polynomial time, as eventually the discrete logarithm computations will become too expensive. One extra benefit is that the class group will not have smooth degree as the maximal order has non-smooth discriminant and the prime factors of the conductor are not special primes.

One difficulty in using non-maximal quadratic orders of large class number is that is must be feasible to generate an oriented curve. This can be done in polynomial time but is extremely costly in practice, even for 1000-bit primes. Our new idea to make this construction more practical is as follows. Assume that we want to generate a curve oriented by $\mathfrak{O} = \mathbb{Z}[\omega]$. First, we find a smooth positive integer g such that $\mathbb{Z}[g\omega]$ embeds into $\mathrm{End}(E_0)$ for the curve $E_0 : y^2 = x^3 + x$. This involves solving a relatively simple Diophantine equation and heuristically every quadratic order $\mathfrak{O}$ embeds into $\mathrm{End}(E_0)$ if it embeds into $B_{p,\infty}$ and its discriminant is of size $\mathrm{disc}(\mathfrak{O}) \gg p^2$. Using the efficient representation of $\mathrm{End}(E_0)$, one only needs to compute an ascending g-isogeny to arrive at a curve oriented by $\mathfrak{O}$. The efficiency gain comes from the fact that previous algorithms computed the orientation on the quaternion side first and executed a full maximal order to elliptic curve algorithm. Note that our technique could be interesting on its own or for further variants of SCALLOP.

This paper is structured as follows. In Sect. 2 we recall some necessary mathematical preliminaries and the high-level idea and design choices of SCALLOP [39]. In Sect. 3 we present our new framework and propose algorithms for generating parameters. In Sect. 4 we discuss the concrete instantiation, implementation challenges and our novel algorithm for generating suitable oriented curve (Algorithm 1).

2 Preliminaries

In this section, we recall the main theoretical concepts needed for understanding SCALLOP, and the class group computation.

2.1 Supersingular Elliptic Curves and Orientations

We begin with a brief review of the required background material on elliptic curves and their orientations. For details, we refer the reader to [39] and the sources cited therein.

Let $p \geq 5$ be a prime and $\overline{\mathbb{F}}_p$ an algebraically closed field of characteristic p. For any elliptic curve $E/\overline{\mathbb{F}}_p$ and any non-negative integer n, we denote by $E[n]$ the group of n-torsion points on E, i.e. the kernel of the multiplication-by-n map on E. Throughout, we will only consider *supersingular* elliptic curves, i.e. curves $E/\overline{\mathbb{F}}_p$ for which $E[p]$ is trivial. Since every supersingular elliptic curve is isomorphic to a curve defined over $\mathbb{F}_{p^2}$, we may assume that E is given by a short Weierstrass equation

$$E : y^2 = x^3 + Ax + B$$

with $A, B \in \mathbb{F}_{p^2}$.

For any isogeny $\phi : E \to E'$ from E to another elliptic curve E', let $\hat{\phi}$ denote its dual and $\deg(\phi)$ its degree. All isogenies herein are assumed to be separable; in particular, $p \nmid \deg(\phi)$ and $\deg(\phi) = \#\ker(\phi)$ is the cardinality of the kernel of ϕ. The only exception is the p-power Frobenius isogeny $\pi : E \to E^p$ defined via $\pi((x, y)) = (x^p, y^p)$, where E^p is given by $y^2 = x^3 + A^p x + B^p$.

Let $\mathrm{End}(E)$ denote the endomorphism ring of E and $\mathrm{End}^0(E) = \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$ the associated endomorphism algebra. Then $\mathrm{End}^0(E) \cong B_{p,\infty}$, the rational quaternion algebra ramified only at p and ∞, and $\mathrm{End}(E)$ is isomomorphic to a maximal order of $B_{p,\infty}$.

Let K be an imaginary quadratic field such that p does not split in K. Then K embeds into $B_{p,\infty}$. A *K-orientation* of E is a (necessarily injective) ring homomorphism $\iota : K \to \mathrm{End}^0(E)$. If $\phi : E \to E'$ is an isogeny, then ϕ induces a K-orientation ι' of E' defined via

$$\iota'(\beta) = \frac{1}{\deg(\phi)}\, \phi \circ \iota(\beta) \circ \hat{\phi} \quad \text{for all } \beta \in K.$$

If there exists an order $\mathfrak{O} \subset K$ (which is unique in this case) such that $\iota(\mathfrak{O}) = \mathrm{End}(E) \cap \iota(K)$, then ι is said to be an *$\mathfrak{O}$-orientation*.[1] Then E is said to be *$\mathfrak{O}$-orientable* and the pair (E, ι) is referred to as an *$\mathfrak{O}$-oriented* elliptic curve.

Note that every $\mathfrak{O}$-orientation ι of E gives rise to an orientation on $E^p = \pi(E)$, since $\mathrm{End}(E) \cong \mathrm{End}(E^p)$. The set $S_{\mathfrak{O}}(p)$ of $\mathfrak{O}$-oriented elliptic curves up

[1] In some sources, $\mathfrak{O}$-orientations are referred to as *primitive* $\mathfrak{O}$-orientations (with $\mathfrak{O}$-orientations without this attribute only requiring $\iota(\mathfrak{O}) \subseteq \mathrm{End}(E) \cap \iota(K)$), or as *optimal embeddings*.

to isomorphism and Frobenius conjugacy is non-empty if and only if p does not divide the conductor of $\mathfrak{O}$. If in addition p splits in K, then $\#S_{\mathfrak{O}}(p) = h(\mathfrak{O})$, the class number of $\mathfrak{O}$. In this case, the class group $\mathrm{Cl}(\mathfrak{O})$ acts freely and transitively on $S_{\mathfrak{O}}(p)$ as follows. For an $\mathfrak{O}$-oriented curve (E, ι_E) and an $\mathfrak{O}$-ideal $\mathfrak{a}$ coprime to the conductor of $\mathfrak{O}$, put $E[\mathfrak{a}] = \bigcap_{\alpha\in\mathfrak{a}} \ker(\iota_E(\alpha))$ and let $\varphi_{\mathfrak{a}}^E : E \to E/E[\mathfrak{a}]$ be the isogeny with kernel $E[\mathfrak{a}]$, of degree $N(\mathfrak{a})$ where $N(\mathfrak{a}) = [\mathfrak{O} : \mathfrak{a}]$ is the norm of $\mathfrak{a}$. Then $\mathfrak{a} \star (E, \iota_E) = (E_{\mathfrak{a}}, \iota_{\mathfrak{a}}) \in S_{\mathfrak{O}}(p)$, where

$$E_{\mathfrak{a}} = E/E[\mathfrak{a}], \qquad \iota_{\mathfrak{a}}(\beta) = \frac{1}{N(\mathfrak{a})}\, \varphi_{\mathfrak{a}}^E \circ \iota_E(\beta) \circ \hat{\varphi}_{\mathfrak{a}}^E \quad \text{for all } \beta \in K.$$

Since principal ideals act trivially on $S_{\mathfrak{O}}(p)$, this extends to an action $\star : \mathrm{Cl}(\mathfrak{O}) \times S_{\mathfrak{O}}(p) \to S_{\mathfrak{O}}(p)$. In practice, $\varphi_{\mathfrak{a}}^E$ will always be given as a product of low degree isogenies of coprime degrees, corresponding to a factorization of $\mathfrak{a}$ into powers of prime ideals in $\mathfrak{O} = \mathbb{Z}[\omega]$. Specifically, if $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$, where $\mathfrak{b}$, $\mathfrak{c}$ are $\mathfrak{O}$-ideals whose norms are relatively prime to each other and to the conductor of $\mathfrak{O}$, then $E_{\mathfrak{b}}[\mathfrak{c}] = \varphi_{\mathfrak{b}}^E(E[\mathfrak{c}])$ and $\varphi_{\mathfrak{a}}^E = \varphi_{\mathfrak{c}}^{E_{\mathfrak{b}}} \circ \varphi_{\mathfrak{b}}^E$. If $\mathfrak{c}$ is primitive, i.e. not divisible by any rational integers other than ± 1, and given by a $\mathbb{Z}$-basis $\{c, u+\omega\}$ with $c = N(\mathfrak{c})$, then $E[\mathfrak{c}]$ is a cyclic group, computable as $E[\mathfrak{c}] = E[c] \cap \ker(\iota_E(\omega) + [u])$, where $[u]$ is the multiplication-by-u map on E.

2.2 SCALLOP

In this section we describe the main mechanism and design choices of SCALLOP [39]. As explained in the previous section, every $\mathfrak{O}$-orientation yields an action of $\mathrm{Cl}(\mathfrak{O})$ on the set of $\mathfrak{O}$-oriented supersingular elliptic curves. The aim of SCALLOP was to find an orientation with the following properties:

1. The class number of $\mathfrak{O}$ is easy to compute;
2. The relation lattice of $\mathrm{Cl}(\mathfrak{O})$ is easier to compute than in the one used in CSIDH (for the same security level);
3. Computing the endomorphism ring of an $\mathfrak{O}$-oriented curve (even when the orientation is provided) is hard (i.e., there does not exists a quantum polynomial-time algorithm for computing the endomorphism ring).

In order to satisfy the first condition, SCALLOP uses non-maximal orders of quadratic fields with small class number. Assuming the factorization of the conductor is known, class numbers of such orders are easy to compute using the formula

$$h(\mathfrak{O}) = \frac{h(\mathfrak{O}_K) f}{[\mathfrak{O}_K^* : \mathfrak{O}^*]} \prod_{q|f} \left(1 - \left(\frac{d_K}{q}\right) \frac{1}{q}\right);$$

see [28, Theorem 7.24]. Here, $\mathfrak{O}_K$ is the maximal order of the field of fractions K of $\mathfrak{O}$, d_K is its discriminant, $\mathfrak{O}_K^*$ and $\mathfrak{O}^*$ are the respective unit groups of $\mathfrak{O}_K$ and $\mathfrak{O}$, f is the conductor of $\mathfrak{O}$, and the product runs over all primes dividing f.

The third condition is somewhat trickier to satisfy in this case. As a particular example, let $\mathfrak{O}$ be an order in $\mathbb{Z}[i]$ of smooth conductor f. Given an $\mathfrak{O}$-orientable

elliptic curve E, one can recover the degree f isogeny from E to E_0, where E_0 is the unique curve oriented by $\mathbb{Z}[i]$, in the following fashion. For each prime factor $\ell \mid f$, one can try all of the ℓ-isogenies from E_0, and choose the right one by evaluating the action of an ideal that is trivial in $\mathrm{Cl}(\mathbb{Z}[(f/\ell)i])$, but not in $\mathrm{Cl}(\mathbb{Z}[fi])$, thus stepwise climbing the oriented isogeny vulcamo.

The natural idea to counter this attack is to take f to be non-smooth; in SCALLOP [39], it is taken to be a large prime. Then the attack fails, as the f-torsion of E is defined over a large extension of $\mathbb{F}_{p^2}$ and one cannot evaluate degree f isogenies without knowing the endomorphism ring of E. On the other hand, when taking an order of prime conductor, it is not obvious how to represent the orientation. In SCALLOP, efficiency is ensured by writing the natural generator σ as a linear combination of 1 and θ, where θ is an endomorphism of smooth degree. Choosing the orientation first and θ afterwards is generally a challenging task in practice. The key idea in SCALLOP is to choose θ first and the corresponding f afterwards. One possible choice is to take the first few primes of the form $4m+1$ and represent them as norms of primes $a_k \pm b_k i$ in $\mathbb{Z}[i]$. Then one can take a particular choice for each prime (either plus or minus) and take their product. If the coefficient of i of this product is prime, then it is an appropriate choice for f.

This motivates a hard problem underlying SCALLOP:

Problem 2.1. Let $\phi : E_0 \to E$ a degree f isogeny. Suppose we can evaluate $\sigma = \phi \circ [i] \circ \hat{\phi} \in \mathrm{End}(E)$ at any point on E (the cost of the evaluation is the size of the representation of the point). Compute $\mathrm{End}(E)$.

In fact, the recent break of pSIDH [23] implies that it is sufficient to be able to evaluate ϕ at any point on E_0, instead of only σ, as then $\mathrm{End}(E)$ can be computed in quantum polynomial time.

These design choices already satisfy the first and third requirement, but in general computing the relation lattice of the class group can be still time consuming. The way this is handled in SCALLOP is to ensure that $f-1$ or $f+1$ is smooth, in which case the relation lattice can be computed by solving low-order discrete logarithms using the Pohlig-Hellman algorithm.

A different route is taken in SCALLOP-HD [24]. There, the authors represent orientations using higher dimensional isogenies. In that setting, f can be selected before choosing θ and then a natural choice is to take $f-1$ to be a product of large powers of 2 and 3.

Finally, we emphasize that in all cases the group action evaluation also entails transporting the orientations (this is not needed in CSIDH as Frobenius provides a canonical orientation by $\mathbb{Z}[\sqrt{-p}]$). In SCALLOP, this requires translating the smooth degree endomorphism θ. When the isogeny degree (corresponding to a small norm ideal) and $\deg(\theta)$ are coprime, this just entails pushing the kernel of θ through the isogeny. The parameter choices in SCALLOP require the translation through non-coprime degree isogenies. This is more complicated and time consuming; we refer the reader to [39, Section 5.2.] for details.

2.3 Class Group Computation

Once an $\mathfrak{O}$-orientation of a curve E is known, the cost of calculating the action of an ideal $\mathfrak{a}$ of large norm on the isomorphism class of E can be greatly reduced by finding prime ideals $\mathfrak{p}_i$ of small norm and exponents x_i such that $\mathfrak{p}_1^{x_1} \dots \mathfrak{p}_k^{x_k} = (\alpha)\mathfrak{a}$ for some $\alpha \in K$. This means that the $\mathfrak{a}$ and $\mathfrak{p}_1^{x_1} \dots \mathfrak{p}_k^{x_k}$ represent the same class in $\mathrm{Cl}(\mathfrak{O})$. Thus, the action of $\mathfrak{a}$ is simply the composition of the actions of the $\mathfrak{p}_i$, which are significantly easier to compute.

Under the Generalized Riemann Hypothesis (GRH), the class group of an order $\mathfrak{O}$ in a number field is generated by the classes of prime ideals of norm less than $48 \log^2(|\Delta_{\mathfrak{O}}|)$ where $\Delta_{\mathfrak{O}}$ is the discriminant of $\mathfrak{O}$ (a direct consequence of [6, Th. 4]; see also [13]). In practice [14], it was observed that significantly fewer primes are necessary to generate $\mathrm{Cl}(\mathfrak{O})$. Once generators $\mathfrak{p}_1, \dots, \mathfrak{p}_k$ of $\mathrm{Cl}(\mathfrak{O})$ are chosen, our goal to minimize the cost of evaluating the action of $\mathfrak{a}$ is to find the smallest exponents $x_1, \dots, x_k$ such that the class $[\mathfrak{a}]$ of $\mathfrak{a}$ in $\mathrm{Cl}(\mathfrak{O})$ is equal to $\prod_i [\mathfrak{p}_i]^{x_i}$. To this effect, we note that the exponent vectors $(e_1, \dots, e_k)$ such that $\prod_i [\mathfrak{p}_i]^{e_i} = [1]$ form a Euclidean lattice $\mathcal{L}$ dubbed the *lattice of relations*. Given an initial decomposition of $[\mathfrak{a}]$ with exponent vector $\boldsymbol{x} = (x_1, \dots, x_k)$, we can obtain a shorter one by finding a vector $\boldsymbol{u} \in \mathcal{L}$ close to $\boldsymbol{x}$. Then $\boldsymbol{x} - \boldsymbol{u}$ is a new exponent vector of such a decomposition of $[\mathfrak{a}]$. If $\boldsymbol{u}$ is the closest vector to $\boldsymbol{x}$, then it yields the shortest decomposition possible.

The typical strategy for decomposing $[\mathfrak{a}]$ with respect to a small set of prime generators $(\mathfrak{p}_i)_{i \le k}$ of $\mathrm{Cl}(\mathfrak{O})$ is to multiply $\mathfrak{a}$ by random short products of the $\mathfrak{p}_i$ and use an ideal reduction technique to obtain $\mathfrak{a}'$ of norm in $O(\sqrt{|\Delta_{\mathfrak{O}}|})$ such that $[\mathfrak{a}'] = [\mathfrak{a}] \prod_i [\mathfrak{p}_i]^{x_i}$ until $\mathfrak{a}'$ is a product of the $(\mathfrak{p}_i)_{i \le k}$ (see for example [13, Alg. 2,3]). A similar strategy can be used to compute a generating set of the lattice of relations $\mathcal{L}$: we look for sufficiently many different random decompositions of $\mathfrak{a} = (1)$. When $\mathfrak{O}$ is the maximal order of K (or is non-maximal with a small conductor), the above strategy is the best known technique. For example, this is the case with the signature scheme CSI-FiSh [20] which requires the fast decomposition of random elements in $\mathrm{Cl}(\mathfrak{O})$ to avoid having to use an expensive *rejection sampling* method to ensure security. The best known technique for computing the lattice of relations between a generating set of primes of such an order $\mathfrak{O}$ relies on the class group computation algorithm of Hafner-McCurley [40]. Under the GRH, its complexity is in $L_{|\Delta|}(1/2)$ where

$$L_x(\alpha) = \exp(O((\log x)^{\alpha}(\log \log x)^{1-\alpha})).$$

For objects of size $\log x$, a complexity in $L_x(0)$ means polynomial time, and a complexity in $L_x(1)$ means exponential time. The subexponential nature of the complexity of the Hafner–McCurley algorithm means that for large values of $|\Delta|$, the search for the relation lattice (and decompositions in $\mathrm{Cl}(\mathfrak{O})$) quickly becomes impractical. Practically speaking, the record computation performed to instantiate CSI-FiSh reached Δ with 512 bits [20].

When the conductor f of the non-maximal order $\mathfrak{O}$ is large, one can significantly reduce the cost of computing the lattice of relations and of ideal decom-

position in $\mathrm{Cl}(\mathfrak{O})$ by using an algorithm due to Klüners and Pauli [44]. From a high level standpoint, this approach takes advantage of the exact sequence

$$1 \to \mathfrak{O}^* \to \mathfrak{O}_K^* \to \bigoplus_{\mathfrak{p}|f} \mathfrak{O}_{K,\mathfrak{p}}^*/\mathfrak{O}_{\mathfrak{p}}^* \to \mathrm{Cl}(\mathfrak{O}) \to \mathrm{Cl}(\mathfrak{O}_K) \to 1,$$

where $\mathfrak{O}_{\mathfrak{p}}$ denotes the localization of $\mathfrak{O}$ at $\mathfrak{p}$. In a nutshell, this means that ideal decomposition (and the search for relations) in $\mathrm{Cl}(\mathfrak{O})$ reduces to ideal decomposition in $\mathrm{Cl}(\mathfrak{O}_K)$ and to the resolution of the Discrete Logarithm Problem (DLP) in the multiplicative groups of the residue fields $\mathfrak{O}_K/\mathfrak{p}$ for $\mathfrak{p} \mid f$ (assuming the factorization of the conductor f is known). See [14, Algorithms 2 and 3] for more details. Note that for a split prime $\mathfrak{p}$, the corresponding instance of the DLP is in a prime field of size $p = \mathcal{N}(\mathfrak{p})$, while in an inert prime, the size of the field is p^2. In the setting of SCALLOP [39], we have $\mathfrak{O}_K = \mathbb{Z}[i]$, which makes all computations in $\mathrm{Cl}(\mathfrak{O}_K)$ easy. On the other hand, no practical implementation beyond 1024-bit discriminants has been achieved due to the hardness of the discrete logarithms. The best known algorithms for solving instances of the DLP are variants of the number field sieve (NFS), which has complexity $L_q(1/3)$, where q is the cardinality of the residue field [43]. Practically speaking, we will only use prime fields, where record computations reach q with approximately 800 bits [18].

In summary, computing the class group of an order of discriminant $\Delta = -df^2$ where $-d$ is a fundamental discriminant and the factorisation of f is known can be achieved in time

$$L_d(1/2) + \sum_{p|f} L_p(1/3).$$

3 New Parameter and Design Choices

In this section, we propose new instantiations of SCALLOP focusing on both security and efficiency.

The key idea is twofold. Firstly, we use a maximal order with larger class number. Secondly, we choose a conductor f that is not smooth but is also not prime. This approach targets concrete efficiency of protocols with security levels equivalent to CSIDH-1024, CSIDH-2048 and CSIDH-4096. Computing class groups of this size in the CSIDH setting is far out of reach with current classical algorithms and infrastructures. SCALLOP was instantiated for the CSIDH-1024 equivalent case [39], but for the higher security levels, finding a conductor f such that $f \pm 1$ is sufficiently smooth might be more challenging. Furthermore, our goal is to provide more efficient group action evaluations.

Instead of the setting of the Gaussian integers, we start with a quadratic order $\mathbb{Z}[\sqrt{-d}]$ where $d > 0$ is a 256-bit integer that will be determined by suitable parameter choices. Computing class groups of this size is feasible in practice [12]. We wish to choose f in such a way that the discriminant of $\mathbb{Z}[f\sqrt{-d}]$ has 1024/2048/4096 bits. This implies that f should have 384/896/1920 bits.

The high-level idea is as follows. We do not fix d right away, but rather restrict our search to maximal orders that contain an endomorphism with particularly smooth degree of the form $2N^2$. This is ensured by introducing a variable parameter a and looking for pairs a, d such that $a^2 + d = 2N^2$. This quantity is then the norm of the element $a + \sqrt{-d} \in \mathbb{Z}[\sqrt{-d}]$. Next, we look for small powers of this element such that the coefficient f of $\sqrt{-d}$ in this power has a particular factorization, which we explain in this section. We focus on the parameters sets corresponding to CSIDH-1024 and CSIDH-2048.

3.1 Effective Orientation from a Generator of a Suborder

Our generation procedure produces parameters f, d such that we know an element $\omega \in \mathbb{Z}[f\sqrt{-d}]$ of smooth norm, which will correspond to the effective orientation. However, the element ω will in fact never be a generator of $\mathbb{Z}[f\sqrt{-d}]$. Instead it will generate a suborder $\mathbb{Z}[\omega] \subset \mathbb{Z}[f\sqrt{-d}]$, with relative index $g = [\mathbb{Z}[f\sqrt{-d}] : \mathbb{Z}[\omega]]$. Therefore, being able to evaluate ω will not satisfy the original definition of an effective representation [39]. However, Proposition 3.1, shows that this causes no extra problems, as long as we can avoid ideals above primes dividing g.

Proposition 3.1. *Let $\mathfrak{O}$ be an imaginary quadratic order, and let $\mathfrak{O}' \subset \mathfrak{O}$ be a suborder of relative index $g = [\mathfrak{O} : \mathfrak{O}']$. Then given an oriented curve $(E, \iota_E) \in S_{\mathfrak{O}}(p)$, together with an endomorphism ω of E generating $\iota_E(\mathfrak{O}') \subset \mathrm{End}(E)$, one can efficiently evaluate the action of any $\mathfrak{O}$-ideal $\mathfrak{l}$ above $\ell \in O(\log(p))$ on (E, ι_E), provided* $\gcd(\ell, g) = 1$.

Proof. Let $\mathfrak{O} = \mathbb{Z}[\delta]$, and let $\mathfrak{l}$ be an $\mathfrak{O}$ ideal of norm ℓ. Recall that finding the isogeny corresponding to $\mathfrak{l} = (a+\delta, \ell)$ is done by computing $E[[a]+\iota_E(\delta)] \cap E[\ell] = (\widehat{[a]+\iota_E(\delta)})(E[\ell])$. To compute this quantity only with knowledge of the isogeny corresponding to ω, we use that $g\delta \in \mathfrak{O}'$, hence $g\delta = c + \omega$ for some $c \in \mathbb{Z}$. Then, since $\gcd(\ell, g) = 1$, we have $\mathfrak{l} = (a + \delta, \ell) = (g(a + \delta), \ell) = (ga + g\delta, \ell) = (ga + c + \omega, \ell)$, and the isogeny corresponding to $\mathfrak{l}$ can be found in the same way as before, given only the evaluation of ω on $E[\ell]$.

For our application in SCALLOP, it will be sufficient to avoid using ideals above primes dividing $g = [\mathbb{Z}[f\sqrt{-d}] : \mathbb{Z}[\omega]]$ in the basis of the lattice of relations, or ignoring the issue entirely, by additionally searching until all small primes dividing g are non-split in $\mathbb{Z}[\sqrt{-d}]$.

3.2 The CSIDH-1024 Case

In this setting we aim to obtain f as the product of three primes of 128 bits each and a very small cofactor. Since we wish to achieve 128-bit security, the natural attacks will fail just as they do when f is prime; thus, there is no compelling reason to take f to be prime. The benefit of this approach is that computing the relation lattice reduces to relatively small finite field discrete logarithm problems.

Fix $N = 2^{129}$; we wish to find a and d such that $d + a^2 = 2N^2 = 2^{259}$. A natural idea would be to take a uniformly at random, compute d accordingly, and raise $a + \sqrt{-d}$ to a small power, hoping that the coefficient of $\sqrt{-d}$ is the product of three prime numbers of size roughly 128 bits (and a possibly very small cofactor). Numbers that are the product of three prime numbers of equal size are relatively dense, but detecting them in practice is potentially hard and time consuming. Instead, we take the more formal approach of computing powers of $a + \sqrt{-d}$ symbolically and expressing the coefficient of $\sqrt{-d}$ in terms of a and d.

For this specific setting the fourth power, $(a + \sqrt{-d})^4$, represents a particularly suitable choice, as the coefficient of $\sqrt{-d}$ in this quantity is

$$4(a^2 - d)a = 4(2a^2 - 2N^2)a = 8(a - N)(a + N)a,$$

which already splits into three factors and the small cofactor 8. So our goal is find d and a subject to the following two restrictions:

- $a, N - a, a + N$ are all small multiples of 128-bit primes;
- The 128-bit prime factors are all split in $\mathbb{Q}(\sqrt{-d})$.

Remark 3.2. The reason for considering $N - a$ instead of $a - N$ is that $N > a$ as a is chosen to be a 128-bit integer and $N = 2^{129}$.

Remark 3.3. The small cofactor (e.g. 8 in the 1024-parameter case) is not problematic, as we are working with curves oriented by the conductor that are simply the product of the larger primes, even if we only have a generator for this suborder. See Sect. 3.1 for details.

The second condition comes from the fact that we need discrete logarithm computations modulo the prime ideals above those primes (see Sect. 2.3) which should involve 128-bit (as opposed to 256-bit) discrete logarithm computations.

The goal is to sample a from a certain residue class to ensure that whenever $a, a + N, a - N$ are (almost) prime (a precise statement is given in Lemma 3.4), then they are also split in $\mathbb{Q}(\sqrt{-d})$. In our specific setting, have $m = 64$.

Lemma 3.4. *Let $N = 2^{2m+1}$ with $m \geq 0$ and $d = 2N^2 - a^2$ with $0 < a < N$. If $a \equiv 19 \pmod{24}$, then whenever a, $(a + N)/3$ and $N - a$ are prime numbers, they split in $\mathbb{Q}(\sqrt{-d})$*

Proof. Recall that a prime q is split in $\mathbb{Q}(\sqrt{-d})$ if and only if $(\frac{-d}{q}) = 1$, where $(\frac{-d}{q})$ denotes the Legendre symbol. Also note that $a \equiv 19 \pmod{24}$ is equivalent to $a \equiv 1 \pmod{3}$ and $a \equiv 3 \pmod{8}$.

Since $a \equiv 3 \pmod{8}$, we have

$$\left(\frac{-d}{a}\right) = \left(\frac{a^2 - 2N^2}{a}\right) = \left(\frac{-2}{a}\right) = \left(\frac{-1}{a}\right)\left(\frac{2}{a}\right) = (-1)(-1) = 1.$$

Similarly, $N - a \equiv -a \equiv 1 \pmod 4$ implies

$$\left(\frac{-d}{N-a}\right) = \left(\frac{(a+N)(a-N) - N^2}{N-a}\right) = \left(\frac{-1}{N-a}\right) = 1.$$

Finally, since $N \equiv 2 \pmod 3$ and $a \equiv 1 \pmod 3$, we see that $a + N$ is divisible by 3. Since $a + N \equiv a \equiv 3 \pmod 4$, we see that $(N + a)/3 \equiv 1 \pmod 4$, so

$$\left(\frac{-d}{(a+N)/3}\right) = \left(\frac{(a+N)(a-N) - N^2}{(a+N)/3}\right) = \left(\frac{-1}{(a+N)/3}\right) = 1.$$

Remark 3.5. Analogous reasoning to the proof of Lemma 3.4 shows that if N is an even power of 2 and $a \equiv 11 \pmod{24}$, then a, $(a + N)/3$ and $N - a$ split again in $\mathbb{Q}(\sqrt{-d})$ when they are prime.

Appropriate parameters can now be generated as follows:

- Set $N = 2^{129}$.
- Sample a random 128-bit number $a \equiv 19 \pmod{24}$.
- Check if a, $(a + N)/3$ and $N - a$ are prime numbers.
- If yes, then set $f = 8(a + N)(N - a)a$ and $d = 2N^2 - a^2$.

3.3 The CSIDH-2048 and CSIDH-4096 Cases

For the larger security levels, using the same method would require the following discrete logarithm computations:

- In the 2048-bit case the maximal order has discriminant 256 bits, so the large prime factors of the conductor will have $\frac{2048-256}{6} \approx 299$ bits.
- In the 4096-bit case, the same calculation gives 640 bits.

In order to save on discrete logarithm computations we will instead use a slight variation of the previous approach. Rather than taking the fourth power of $a + \sqrt{-d}$, we take the the 12^{th} power and again consider the coefficient of $\sqrt{-d}$, which is given by the expression

$$4a(a^2 - d)(a^2 - 3d)(3a^2 - d)(a^4 - 14a^2d + d^2).$$

Again we let $N = 2^{2m+1}$ be an odd power of 2 of appropriate size and search for d of the form $d = 2N^2 - a^2$. Then the factorization of the expression above becomes

$$128a(a+N)(a-N)(2a^2 - 3N^2)(2a^2 - N^2)(2a^2 - 2aN - N^2)(2a^2 + 2aN - N^2).$$

Note that the conductor now has 7 large factors, of varying size, though we will see that by choice of N and a, the smallest will still be 128-bit.

Lemma 3.6. *Let $N = 2^{2m+1}$ with $m \geq 0$ and $d = 2N^2 - a^2$ with $0 < a < N/\sqrt{2}$. Let*

$$P = \frac{N^2}{2} - a^2, \quad Q = \frac{3N^2 - 2a^2}{10}.$$

If $a^2 \equiv 1 \pmod{30}$, then whenever P and Q are prime numbers, they split in $\mathbb{Q}(\sqrt{-d})$.

Proof. Note that $a^2 \equiv 1 \pmod{30}$ if and only if a is odd and a^2 is congruent to 1 modulo both 3 and 5. The first of these properties is equivalent to $a^2 \equiv 1 \pmod 8$ (so we actually obtain $a^2 \equiv 1 \pmod{120}$).

Clearly P is an integer. Since $3N^2 \equiv 2 \pmod{10}$ and $a^2 \equiv 1 \pmod{10}$, Q is also an integer. Since $a^2 < N^2/2$, we see that both P and Q are positive.

We have $-d = -3N^2/2 - P = -6 \cdot (2^{2m})^2 - P$, so

$$\left(\frac{-d}{P}\right) = \left(\frac{-1}{P}\right)\left(\frac{2}{P}\right)\left(\frac{3}{P}\right).$$

Now a is odd, so $P \equiv -a^2 \equiv -1 \pmod 8$ (and hence also $P \equiv -1 \pmod 4$). Furthermore, $N^2/2 \equiv 2 \pmod 3$ and $a^2 \equiv 1 \pmod 3$ imply $P \equiv 1 \pmod 3$. It follows that

$$\left(\frac{-1}{P}\right) = -1, \quad \left(\frac{2}{P}\right) = 1, \quad \left(\frac{3}{P}\right) = -\left(\frac{P}{3}\right) = \left(\frac{1}{3}\right) = -1,$$

so $\left(\frac{-d}{P}\right) = 1$. Similarly, $-d = -N^2/2 - 5Q = -2 \cdot (2^{2m})^2 - 5Q$. We have $5Q \equiv -a^2 \equiv -1 \pmod 8$, and hence $Q \equiv 3 \pmod 8$. It follows that

$$\left(\frac{-d}{Q}\right) = \left(\frac{-2}{Q}\right) = \left(\frac{-1}{Q}\right)\left(\frac{2}{Q}\right) = (-1)(-1) = 1.$$

Lemma 3.7. *Let $N = 2^{2m+1}$ with $m \geq 0$ and $d = 2N^2 - a^2$ with $(\sqrt{3}-1)N/2 < a < N$. Let*

$$R = a^2 + aN - \frac{N^2}{2}, \quad S = \frac{N^2/2 + aN - a^2}{3}.$$

If $a \equiv 7 \pmod{12}$, then whenever R and S are prime numbers, they split in $\mathbb{Q}(\sqrt{-d})$.

Proof. The congruence condition on a yields $a \equiv 3 \pmod 4$ and $a \equiv 1 \pmod 3$.

Clearly R is an integer, and since $a \equiv 1 \pmod 3$ and $N \equiv 2 \pmod 3$, we see that S is also an integer.

We have $R = (a + N/2)^2 - 3N^2/4$ which is positive because of the lower bound on a. Moreover, $S > N^2/6 > 0$ as $a < N$.

Now $-d = R - N(a + 3N/2) = R - 2 \cdot (2^m)^2(a + 3N/2)$, so

$$\left(\frac{-d}{R}\right) = \left(\frac{-1}{R}\right)\left(\frac{2}{R}\right)\left(\frac{a + 3N/2}{R}\right).$$

Since $R \equiv a^2 \equiv 1 \pmod 8$, we have

$$\left(\frac{-1}{R}\right) = \left(\frac{2}{R}\right) = 1, \quad \left(\frac{a+3N/2}{R}\right) = \left(\frac{R}{a+3N/2}\right).$$

It is easy to verify that $R = (a+3N/2)(a-N/2) + N^2/4$, so $\left(\frac{R}{a+3N/2}\right) = 1$. Hence $\left(\frac{-d}{R}\right) = 1$.

Similarly, $-d = -N(3N/2-a) - 3S = -2 \cdot (2^m)^2(3N/2-a) - 3S$, where we note that $3N/2 - a > 0$ as $a < N$. So

$$\left(\frac{-d}{S}\right) = \left(\frac{-1}{S}\right)\left(\frac{2}{S}\right)\left(\frac{3N/2-a}{S}\right).$$

Since $3S \equiv -a^2 \equiv -1 \pmod 8$, we have $S \equiv -3 \pmod 8$, so $S \equiv 1 \pmod 4$ and we obtain

$$\left(\frac{-1}{S}\right) = 1, \quad \left(\frac{2}{S}\right) = -1, \quad \left(\frac{3N/2-a}{S}\right) = \left(\frac{S}{3N/2-a}\right),$$

and hence $\left(\frac{-d}{S}\right) = -\left(\frac{S}{3N/2-a}\right)$.

Again one readily checks that $3S = (3N/2-a)(a+N/2) - N^2/4$. Since $3N/2 - a \equiv -a \equiv 1 \pmod 4$, we have

$$\left(\frac{3S}{3N/2-a}\right) = \left(\frac{-1}{3N/2-a}\right) = 1$$

and

$$\left(\frac{3}{3N/2-a}\right) = \left(\frac{3N/2-a}{3}\right) = \left(\frac{-a}{3}\right) = \left(\frac{-1}{3}\right) = -1.$$

Overall, we get

$$\left(\frac{-d}{S}\right) = -\left(\frac{S}{3N/2-a}\right) = -\left(\frac{3S}{3N/2-a}\right)\left(\frac{3}{3N/2-a}\right) = -1(-1) = 1.$$

The numerical values of the constants in the bounds on a relative to N appearing in Lemmas 3.6 and 3.7 are $1/\sqrt{2} \approx 0.707$ and $(\sqrt{3}-1)/2 \approx 0.336$.

Combining the congruence conditions on a in Lemmas 3.6 and 3.7 yields $a \equiv 19$ or $31 \pmod{60}$. In conjunction with the restriction $a \equiv 19 \pmod{24}$ from Lemma 3.6, we would require $a \equiv 19$ or $91 \pmod{120}$. However, even if any of the smaller factors a, $N-a$ and $(N+1)/3$ are not prime and contain non-split prime factors, the corresponding discrete log computations are negligible compared to the cost of the discrete log extraction modulo P, Q, R and S.

As in the CSIDH-1024 case, we can now generate suitable parameters for CSIDH-2048 and CSIDH-4096 as follows (note that $30720 = 2^{11} \cdot 3 \cdot 5$).

– Set $N = 2^{129}$ for CSIDH-2048 or $N = 2^{175}$ for CSIDH-4096.

- Sample a random number $a \equiv 19$ or $31 \pmod{60}$ with $(\sqrt{3}-1)N/2 < a < N/\sqrt{2}$.
- Check if P, Q, R, S as given in Lemmas 3.6 and 3.7 are prime numbers.
- If yes, then set $f = 30720a(N+a)(N-a)PQRS$ and $d = 2N^2 - a^2$.

Remark 3.8. It might not be immediately obvious why we set $N = 2^{129}$ for CSIDH-2048 or $N = 2^{175}$ for CSIDH-4096. For CSIDH-2048 we make use of Sect. 3.1, i.e., the case when the smooth-norm endomorphism only generates a suborder. In this case we ignore the factor $(a^4 - 14a^2d + d^2)$ for efficiency purposes. Then the size of N is determined by the fact that we need the factor f to have 896 bits. For the CSIDH-4096 case we aim to optimize the precomputation time; hence, we would desire less costly discrete logarithm computations and we utilize an endomorphism that generates the entire order. The size of N is then determined by the fact that we need the factor f to have 1920 bits.

Alternatively, one could also use the same approach for the 4096-bit case as for the 2048-bit case, which would result in larger discrete logarithm computations. However, this approach has the advantage of having significantly fewer trial iterations to ensure that all factors are prime (only 4 factors compared to 7).

Remark 3.9. We have $R = (a + N/2)^2 - 3N^3/4$, $3S = (a - N/2)^2 - 3N^2/4$. So for any prime q such that 3 is a quadratic residue modulo q, say $3 \equiv u^2 \pmod{q}$, we have

$$R \equiv \left(a + (1+u)\frac{N}{2}\right)\left(a + (1-u)\frac{N}{2}\right) \pmod{q},$$

with an analogous factorization for S. Thus, if $a \equiv \pm(1 \pm u)N/2 \pmod{q}$, for all four possible sign combinations, then R or S is a multiple of q. For example, if $a \equiv \pm 1$ or $\pm 7 \pmod{11}$, then one of R, S is divisible by 11. This rules out four residue classes modulo q for a for every prime $q \equiv \pm 1 \pmod{12}$. A test for eliminating these congruence class requires the computation of a square root of 3 (mod q). For small primes q such as 11 and 13, this idea might aid in speeding up the search for suitable parameters, but for large q, such a square root computation is too costly to be useful.

3.4 An Intermediate Case

One drawback (from a purely implementation standpoint) of the 2048-bit method is that it requires computing discrete logarithms in non-prime finite fields. This is not a theoretical obstacle; however, for the apparent sizes, existing open source software is only available for prime fields. Instead, we provide again a slight variation of the above method tuned to an intermediate parameter set where the discriminant of the order is roughly 1500 bits. This is simply based on considering the coefficient of $\sqrt{-d}$ in the 6th power of $a + \sqrt{-d}$, given by

$$2(3a^2 - d)(a^2 - 3d)a.$$

Now we set $d = 4N^2 - a^2$ where $N = 2^{126}$. In this setting the factorization of the quantity above is

$$32 \cdot 3a(a - N)((a + N)/3)(a^2 - 3N^2).$$

Here, our goal is to ensure that the last four terms are all prime numbers which split in $\mathbb{Q}(\sqrt{-d})$. A similar calculation as in Sect. 3.2 shows that if a is chosen to be a prime congruent to 29 modulo 36, then all four factors are indeed split primes. Here, the largest discrete logarithm computation corresponds to the last factor (as it is quadratic). In the 1500-bit case, this amounts to a 500-bit discrete logarithm computation.

3.5 Security

The security offered by these new SCALLOP parameters can be analyzed similarly to the earlier parameter sets, with a few differences taken into consideration for the changes in f and d.

Note that with every parameter set we target 128-bit classical security (and comparable quantum security), as the debate on CSIDH security is about which parameter set achieves 128-bits of security [8,49].

Generic Attacks. Recall that a free and transitive group action $\star$ on a group G and set X creates a *hard homogeneous space* if it can be evaluated efficiently and the following two problems are intractable.

Problem 3.10 (Vectorization). Given $x, y \in X$, find $g \in G$ such that $g \star x = y$.

Problem 3.11 (Parallelization). Given $x, g \star x, h \star y \in X$ (for undisclosed $g, h \in G$), find $(g \cdot h) \star x$.

It is a *very hard homogeneous space* if the following problem is also intractable.

Problem 3.12 (Decisional Parallelisation). Given $x, y, u, v \in X$, decide whether there exists some $g \in G$ satisfying $g \star x = y$ and $g \star u = v$.

When $G = \mathrm{Cl}(\mathfrak{O})$ and $X = S_{\mathfrak{O}}(p)$ we refer to Problem 3.10 as the $\mathfrak{O}$-Vectorization problem; similarly for the other two problems. It is known that $\mathfrak{O}$-Vectorization reduces in quantum polynomial time to $\mathfrak{O}$-Parallelization [57, Theorem 3].

The fastest known generic classical algorithm for solving the $\mathfrak{O}$-Vectorization problem [57, Proposition 3] runs in time

$$\log(p + d)^{O(1)} \min\left(p^{1/2}, f^{1/2}\right),$$

where $d = |\mathrm{disc}(\mathfrak{O})|$. An asymptotically faster quantum attack [57, Proposition 4] on $\mathfrak{O}$-Vectorization utilizing Kuperberg's algorithm for the Abelian hidden shift problem [45] has complexity

$$\log(p)^{O(1)} L_{|\mathrm{disc}(\mathfrak{O})|}(1/2).$$

There are faster quantum algorithms for this problem [21,25,41] that rely on specific group structures; however, the class groups from Sect. 3 have drastically different structures. The general principle in these results is that if the exponent of the group is small, then special purpose algorithms are faster than Kuperberg's algorithm. All SCALLOP variants use class groups with large exponent but our new parameter choice has the extra advantage that the group order is not smooth and has extra flexibility (e.g., one can ensure that the group order has a large prime factor which is useful for threshold schemes [32]).

Remark 3.13. If one does not care about the smoothness of the class group, then we can restrict the parameter search to only search for a divisible by a large power of 2, thereby making most of the discrete logarithm computations significantly easier trivial. However, this is in general not necessary, as the sizes of the discrete logarithms are computationally heavy, but still doable on a laptop.

For the security of the $\mathfrak{O}$-Decisional Parallelization problem, in addition to the above attacks on Vectorization, there are also distinguishers built from quadratic characters [19,22]. These attacks apply to our case as the order of the class group is even by design. However, one can counter this attack in the usual sense by restricting the group action to acting only by elements of $\mathrm{Cl}(\mathbb{Z}[\omega])^2$ (this is only necessary for applications where the decisional problem must be hard). Such characters exist for each divisor $m \mid \mathrm{disc}(\mathfrak{O})$; however, their evaluation (at least classically) takes time polynomial in m. Hence this class of attacks are inefficient when applied to our parameters.

pSIDH Type Attacks. The original SCALLOP construction crucially relies on the hardness of Problem 2.1. The hardness of this problem stems from the following observation. Given the evaluation of $\phi \circ \iota \circ \hat{\phi}$ on a point P, the required task is to find the subgroup generated by $\phi(P)$. If this can be done for arbitrary P, then Problem 2.1 can be efficiently solved [57, Proposition 7].

Let n_P denote the order of P. Then purely working modulo n_P will not be sufficient to recover $\langle \phi(P) \rangle$. Indeed, when precomposing ϕ with an endomorphism that commutes with ι and whose degree is congruent to 1 (mod n_P), the evaluation of the composition does not change, whereas the isogeny ϕ may change. Specifically, for integers a, b satisfying $a^2 + b^2 \equiv 1 \pmod{n_P}$, one obtains

$$\left(\phi \circ (a + b\iota) \circ \iota \circ (a - b\iota) \circ \hat{\phi}\right)(P) = \left((a^2 + b^2)\phi \circ \iota \circ \hat{\phi}\right)(P) = \left(\phi \circ \iota \circ \hat{\phi}\right)(P).$$

There are however certain exceptions.

The following counter-example to the above assertion is based on [17, §10]. As in Problem 2.1, consider the isogeny $\phi : E_0 \to E$ of degree f and an endomorphism $\theta \in \mathrm{End}(E_0)$ of degree n_θ. Suppose we can evaluate some $\sigma = \phi \circ \theta \circ \hat{\phi}$ at any point on E. Note that this is a more general setting than Problem 2.1, since it allows an arbitrary $\theta \in \mathrm{End}(E_0)$. Assume that n_P is prime and coprime to f and n_θ. We show that if the subgroup generated by P is fixed by θ, then this

subgroup can be efficiently computed. That is, given $P \in E_0[n_P]$, we compute $[\lambda]\phi(P)$ for some $\lambda \in (\mathbb{Z}/n_P\mathbb{Z})^*$.

Suppose $\theta(P) = [a]P$; that is, θ fixes the subgroup generated by P. Using the oracle for σ on $E[n_P]$, we can extract discrete logarithms and use linear algebra to compute a point $U \in E[n_P]$ satisfying $\sigma(U) = [fa]U$. We show that $\phi(P)$ is in the subgroup generated by U; by comparing orders, we see that $U = [\lambda]\phi(P)$ for some invertible λ which is our goal. Let $Q \in E_0[n_P]$ be a point independent of P satisfying $\theta(Q) = [b]Q$ for some invertible $b \not\equiv a \pmod{n_P}$.

By coprimality and the independence of P and Q, we can write $U = \phi([x]P + [y]Q)$ for some integers x, y. We show that $y = 0$:

$$\begin{aligned}[fa]U = \sigma(U) &= \sigma \circ \phi\left([x]P + [y]Q\right) \\ &= [f]\phi \circ \theta\left([x]P + [y]Q\right) = [fxa]\phi(P) + [fyb]\phi(Q).\end{aligned}$$

Multiplying both sides by $f^{-1}a^{-1} \pmod{n_P}$ yields $U = [x]\phi(P) + [yba^{-1}]\phi(Q)$. Since $U = \phi([x]P + [y]Q)$ and $ba^{-1} \not\equiv 1 \pmod{n_P}$, we conclude that $y = 0$.

So in this setting, it is possible to glean local information on the evaluation of P. This could potentially be combined with an l-adic approach where local information is combined to obtain the global evaluation of certain points. Note that the above argument is a local one, as precomposing ϕ with an endomorphism changes the degree (which should be the fixed f) globally but not locally. Another potential approach is to utilize the attack [23] on the NIKE scheme pSIDH [46] directly on isogenies of the form $\phi \circ \iota \circ \hat{\phi}$, as these can be evaluated at any point. There is a similar group action on the set of these types of endomorphisms that is rather closely related to the corresponding isogenies ϕ. However, it is not obvious how to evaluate this action as the approach from [23] does not translate.

A crucial component in defining Problem 2.1 is that the class number of the maximal order in SCALLOP is 1. The reason is that [46, Proposition 8] implies that there is a degree f isogeny from any public key curve to the curve $y^2 = x^3 + x$. In our case, we use a maximal order with large class number, where the corresponding statement is no longer effective. There is always a degree f isogeny to a supersingular elliptic curve oriented by the maximal order, but now there are many such curves and it is computationally hard to identify the correct one (since the discriminant is a 256-bit number, one expects roughly 2^{128} possible choices which makes enumeration approaches infeasible). It is important to note that for a key exchange the endomorphism ring of the starting curve can (and should) be known to everyone, which is the case for all isogeny-based group action approaches. The special feature of the original SCALLOP is that an f-isogeny to $y^2 = x^3 + x$ exists not just for the starting curve, but also for the curve obtained after applying the group action. In conclusion, a quantum polynomial-time attack on Problem 2.1 does not immediately provide an attack on PEARL-SCALLOP.

Torsion-Point Attacks. As mentioned before, choosing f to be smooth would be insecure essentially due the torsion-point attack framework pioneered in [51].

The cost of this attack depends on f-isogeny evaluations and representing points of order f. In our case, both are very costly as the f-torsion is defined over an extension of the base field $\mathbb{F}_{p^2}$ of degree larger than 2^{128} and the evaluation of f-isogenies requires at least 2^{64} field operations (utilizing [7]) in this large extension. At present, there seems to be no practical advantage to using a prime degree isogeny instead of an isogeny which is the product of a few primes.

4 Explicit Instantiation of PEARL-SCALLOP

In this section we discuss the implementation details, followed by the timing results comparing our new parameter set with SCALLOP [39]. Our implementation uses SageMath [56], PARI/GP [55], CADO-NFS [54] and NTL [53], and can be found in the repository https://www.github.com/biasse/SCALLOP-params.

4.1 Discriminant Generation

We describe our instantiation of CSIDH-1024. Following the method explained in Sect. 3, we generate a quadratic order $\mathfrak{O}$, providing the class group action, together with an element $\omega \in \mathfrak{O}$, which we will use to evaluate the action. Numerically, reusing the notation from Sect. 3, we set $N = 2^{129}$ and find that the prime

$$a = 340282366920938463463374607431770911081$$

generates the following values of d and f:

$$\begin{aligned} d &= 18466951 \cdot 19397359 \cdot 1148147065021103529892731 53 \\ &\quad \cdot 19707957158568828802463753229623541551, \\ f &= 2^3 \cdot 3 \cdot 340282366920938463463374607431760112581 \\ &\quad \cdot 340282366920938463463374607431770911081 \\ &\quad \cdot 340282366920938463463374607431776310331. \end{aligned}$$

Finding this value of a took seconds on a laptop; similarly, a suitable value of a for CSIDH-2048 was found within minutes.

We then select a value of n and let $\ell_1, \ldots, \ell_n$ be the first n split primes that do not divide the relative conductor $[\mathbb{Z}[\omega] : \mathfrak{O}]$. Subsequently, we choose a prime of the form

$$p = c2^e \prod_{i=1}^{n} \ell_i - 1,$$

where e satisfies $N(\omega) = 2^{2e}$, the ℓ_i correspond to the primes in the basis of the lattice of relations, and c is a small cofactor such that p is a prime satisfying $\left(\frac{-\mathrm{disc}(\mathfrak{O})}{p}\right) = 1$. Continuing our example parameters for CSIDH-1024, we found that $n = 75$, $e = 518$ and $c = 817$ generate suitable parameters for our chosen values of a and N.

4.2 Computing the Relation Lattice

Our implementation of relation lattice computation is a variant of the algorithms mentioned in Sect. 2.3. Let K be an imaginary quadratic field with maximal order $\mathfrak{O}_K$, $f \geq 1$ an integer, and $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_K$ the order of conductor f. Let S be a finite set of prime ideals of $\mathfrak{O}_K$ not dividing f and $S_{\mathfrak{O}} = \{\mathfrak{p} \cap \mathfrak{O} : \mathfrak{p} \in S\}$, so that every ideal in $S_{\mathfrak{O}}$ is invertible. For our purposes, S will contain a set of ideals that are particularly easy to evaluate, i.e. of small norm. Recall that the group $\mathfrak{O}_S^\times$ of S-units of $\mathfrak{O}$ is the set of $u \in K^\times$ such that the ideal $u\mathfrak{O}$ is a product of the elements of $S_{\mathfrak{O}}$. Define the morphism $V_S \colon K^\times \to \mathbb{Z}^S$ by

$$V_S(x) = (v_{\mathfrak{p}}(x))_{\mathfrak{p} \in S},$$

where $v_{\mathfrak{p}}(x)$ denotes the $\mathfrak{p}$-adic valuation of x. We write $V_S^{\mathfrak{O}}$ for the restriction of V_S to $\mathfrak{O}_S^\times$. Then the kernel of $V_S^{\mathfrak{O}}$ is the group of units $\mathfrak{O}^\times$ (which is finite, cyclic, and equal to $\{\pm 1\}$ unless $\mathfrak{O} = \mathbb{Z}[\sqrt{-1}]$ or $\mathfrak{O} = \mathbb{Z}[\sqrt{-3}]$), and the cokernel of $V_S^{\mathfrak{O}}$ is canonically isomorphic to the subgroup of the class group of $\mathfrak{O}$ generated by the classes of elements in $S_{\mathfrak{O}}$. The image of $V_S^{\mathfrak{O}}$ is the relation lattice of $\mathfrak{O}$ (relative to S).

Remark 4.1. From a relation $v \in V_S(\mathfrak{O}_S^\times)$, we can easily recover a preimage: construct the corresponding ideal and apply lattice reduction with respect to the norm. Then the shortest vector will be a generator of the ideal and a preimage of v. Thus, computing the S-unit group is equivalent to computing the relation lattice.

The paper [44] focuses on computing the class group of an order, while we are more interested in the relation lattice, so we use a variant of their algorithm. This has the additional advantage of not requiring any computation of discrete logarithms in the class group, once a relation lattice is known for the maximal order. We use the natural reduction modulo f map $\mathrm{Red}_{S,f} \colon \mathfrak{O}_{K,S}^\times \to (\mathfrak{O}_K/f\mathfrak{O}_K)^\times$, which is well-defined since f is not divisible by any ideals in S. We will compute $\mathfrak{O}_S^\times$ using the well-known identity

$$\mathfrak{O}_S^\times = \mathrm{Red}_{S,f}^{-1}((\mathbb{Z}/f\mathbb{Z})^\times). \tag{1}$$

The Klüners–Pauli algorithm is implemented in Magma [16], but appears to be unable to handle instances with a maximal order of non-trivial discriminant.

With this setup in place, we can describe our implementation of computing the relation lattice relative to a set S.

1. Pick a set S_0 containing S that provably generates the class group of K and compute the relation lattice of $\mathfrak{O}_K$ relative to S_0 using PARI/GP [55]. There are two algorithms implemented for this task: `bnfinit`, which is designed for number fields of arbitrary degree, and `quadclassunit`, which is a faster implementation for quadratic fields. However, none of these implementations uses sieving[2], and even for 256-bit discriminants they struggle. We therefore adapted Pari's implementation (originally created by Papanikolaou and

[2] Our implementation with sieving will be included in the next release of PARI/GP.

Roblot) of the quadratic sieve factoring algorithm (MPQS) so that it computes the relation lattice of $\mathfrak{O}_K$ relative to S_0.
2. Compute the relation lattice of $\mathfrak{O}_K$ relative to S from the relation lattice relative to S_0 by computing an integer kernel.
3. Compute discrete logarithms of a basis of S-units of $\mathfrak{O}_K$ modulo all prime power divisors of f, yielding a description of the map $\mathrm{Red}_{S,f}$ as a matrix. We use
   (a) the Pari implementation of discrete logarithm computations for field sizes up to 150 bits, and
   (b) CADO-NFS for larger sizes.
4. Compute the relation lattice of $\mathfrak{O}$ relative to S using (1) by computing a kernel modulo the exponent of $(\mathfrak{O}_K/f\mathfrak{O}_K)^\times$.
5. Check that the cardinality of the cokernel of $V_S^{\mathfrak{O}}$ is equal to the class number of $\mathfrak{O}$, thus proving that S generates the class group of $\mathfrak{O}$.

Let $K = \mathbb{Q}(\sqrt{-d})$ and let S be the set of prime ideals above $\ell_1, \ldots, \ell_n$. The running times of the various steps were as follows, using a single core of an Intel Xeon CPU E5-2623 v3 @ 3.00 GHz:

Step 1	Step 2	Step 3a	Step 4	Step 5
3 h	38 s	31 min	12 ms	64 ms

The structure of the class group of the maximal order is

$$\mathrm{Cl}(\mathfrak{O}_K) \cong C_{8529112802465676564302495690233842625 6} \times C_2^2.$$

The bottleneck for CSIDH-2048 and CSIDH-4096 will be the computation of the discrete logarithms. The running times of the steps for CSIDH-1536 were as follows (single core as above).

Step 1	Step 2	Step 3a	Step 3b	Step 4	Step 5
3 h	1 h	30 min	1 h	27 ms	140 ms

4.3 Lattice Reduction

We performed lattice reduction of the relation lattice of $\mathfrak{O}$ relative to S on a 64-core AMD Threadripper 3990X 2.9GHz CPU with 256GB of DDR4 RAM. We used the implementation of the BKZ algorithm [50] from the G6K python library [3] originally presented in [2]. For the lattice reduction corresponding to the CSIDH-1024 parameters, the lattice dimension was $|S| = 75$. Then the reduction of an input product $\prod_i \mathfrak{p}^{x_i}$ was obtained by using Babai's nearest plane algorithm [5] to find a lattice point $\boldsymbol{u}$ close to $\boldsymbol{x}$, followed by a random walk approach used in CSI-fish [11] due to Doulgerakis, Laarhoven and de Weger [35]. The timings, given in CPU seconds, were as follows:

Security level	HKZ reduction	Babai & random walk
1024	739 s	3.9 s
1536	750 s	4.6 s

Additionally, we performed weighted reductions of the input lattice to account for the cost of evaluating the action of a prime ideal $\mathfrak{p} \in S$. Indeed, since that cost is proportional to $\mathcal{N}(\mathfrak{p})$, the cost of the evaluation of the product $\prod i\mathfrak{p}_i^{x_i}$ is proportional to $\sum_i x_i \mathcal{N}(\mathfrak{p}_i)$. Therefore, we reduced a weighted lattice where the i-th coordinate is multiplied by $1 + c\mathcal{N}(\mathfrak{p}_i)$, for some constant $0 < c < 1$. This strategy successfully produced short decompositions where the coefficients corresponding to larger primes were significantly smaller than those corresponding to the smaller primes of S. The optimal value of c with respect to the group action evaluation is hard to compute, as it depends on many factors, but it can be estimated for specific implementations.

We can always choose the size of our lattice so that lattice reduction will not be the bottleneck to instantiate our system with CSIDH-2048 and CSIDH-4096 parameters, though note that the relative size of the reduced vectors will be larger at these parameters, as the class group is larger.

4.4 Generating the Starting Curve

Recall that we need a starting curve that is $\mathfrak{O}$-oriented and an efficient way to evaluate the orientation. This is achieved through a triple (E, P, Q) where P, Q are points on E generating smooth isogenies ϕ_P, ϕ_Q, such that their composition $\hat{\phi}_Q \circ \phi_P$ is an element of $\mathfrak{O} \subseteq \mathrm{End}(E)$ (see Sect. 3.1 for a remark on not using a generator).

We now present a new algorithm for efficiently generating a curve with an effective orientation by an order $\mathfrak{O}$, provided an element in $\mathfrak{O}$ of smooth norm is known. Our algorithm is more general, and comparable in efficiency to the curve generation algorithm SetUpCurve from the original SCALLOP paper [39, Algorithm 1], which only works for suborders of a quadratic order that embed into a special p-extremal maximal quaternion order (such as $\mathbb{Z} + f\mathbb{Z}[i]$).

We fix a special p-extremal maximal quaternion order $\mathcal{O}_0$. The idea of the algorithm is to utilize the fact that the quaternion embedding problem is easily solvable in $\mathcal{O}_0$, provided we hit an easy Cornacchia instance [4, Remark 5.14] (see also [37, Proposition 2]). Hence, we can try different smooth values of g until we can compute an embedding of $\mathbb{Z} + g\mathfrak{O}$ into $\mathcal{O}_0$. We use the heuristic algorithm GenericOrderEmbeddingFactorisation for this purpose, see [37, Algorithm 3]. Given such an embedding, it is then easy to compute an ideal corresponding to the ascending isogeny of degree g (since g was chosen to be smooth). The codomain of this isogeny will then be oriented by $\mathfrak{O}$.

Next, to compute the effective orientation given by a smooth element ω, finding the corresponding kernel generators can easily be done by the standard technique of factoring the ideal into two equal parts, and then translating back and forth to E_0. This technique is the same as what is used in SQIsign [31]. The complete method is summarized in Algorithm 1.

Translating a smooth ideal to its corresponding isogeny (or kernel) is in Algorithm 1 denoted by IdealToIsogeny (or IdealToKernel). See, for instance [47, Algorithm 19].

Remark 4.2. Note that GenerateStartingCurve is dual to the original method from SCALLOP in the following sense. In the original method, one computes a *descending* isogeny from a curve with special, p-extremal endomorphism ring oriented by a *superorder*, while in GenerateStartingCurve, we compute the *ascending* isogeny from a curve with special, p-extremal endomorphism ring oriented by a *suborder*.

Algorithm 1. GenerateStartingCurve(γ, p, ω, T)

Input: A generator γ of $\mathfrak{O}$, a prime p such that $\left(\frac{-\mathrm{disc}(\mathfrak{O})}{p}\right) = 1$, an element $\omega \in \mathfrak{O}$ with $N(\omega) = L_1 L_2$, L_i smooth and $E[L_i]$ defined over $\mathbb{F}_{p^2}$, and a powersmooth value $T \gg p/\sqrt{\mathrm{disc}(\mathfrak{O})}$, with $\gcd(L, T) = 1$.
Output: An effectively oriented curve (E, P, Q)

Let i, j, k be a basis of $B_{p,\infty}$, such that $i^2 = -q$ and $j^2 = -p$.
Let $\mathcal{O}_0$ be a special, p-extremal maximal order in $B_{p,\infty}$, and E_0 a supersingular elliptic curve with $\mathrm{End}(E_0) \cong \mathcal{O}_0$.
for $n \mid T$ such that $T/n > p/\sqrt{\mathrm{disc}(\mathfrak{O})}$ **do**
    Set $g := T/n$.
    Set $\delta :=$ GenericOrderEmbeddingFactorisation$(\mathcal{O}_0, \mathrm{trd}(g\gamma), \mathrm{nrd}(g\gamma))$.
    if $\delta \neq \perp$ **then**
        Break loop.
    end if
end for
Set $I := \mathcal{O}_0\langle g, \delta\rangle$.
Compute ϕ_I from I using IdealToIsogeny.
Set $E := \phi_I(E_0)$, and compute $\mathcal{O} := \mathcal{O}_R(I)$.
Let $H_1 := \mathcal{O}\langle L_1, \omega\rangle$, and $H_2 := \mathcal{O}\langle L_2, \bar{\omega}\rangle$.
Translate $[I]^* H_i$ to their kernel generators K_i using IdealToKernel.
Set $P := \phi_I(K_1)$ and $Q = \phi_I(K_2)$.
return (E, P, Q)

Proposition 4.3. GenerateStartingCurve *is correct and runs in probabilistic polynomial time, under standard heuristics.*

Proof. First, we prove that the first part of the algorithm terminates and is correct. Each call to GenericOrderEmbeddingFactorisation runs in polynomial time under [37, Heuristic 1, Heuristic 2], and again under [37, Heuristic 2], the number of maximal orders oriented by $\mathbb{Z} + g\mathfrak{O}$ is $O(p)$, hence a solution is expected to exist.

Assume now that the orientation given by δ has been found. The quaternion ideal $I := \mathcal{O}_0\langle \delta, g\rangle$ corresponds to an ascending isogeny of degree g (it is generated by the unique invertible $(\mathbb{Z} + g\mathfrak{O})$-ideal of norm g). Further, since g is

powersmooth of magnitude $O(p/\sqrt{\text{disc}(\mathfrak{O})})$, translating I to its corresponding isogeny ϕ_I using IdealToIsogeny is efficient.

Once we have a curve E with $\text{End}(E) \cong \mathcal{O}$ oriented by $\mathfrak{O}$, we use the smooth norm ideal $\mathcal{O}\langle\omega\rangle$ to find the effective orientation. This is done by writing $\mathcal{O}\langle\omega\rangle = \bar{H}_2 \cdot H_1$, where H_i can be efficiently translated, by pulling back to an ideal of $\mathcal{O}_0$, using I, since $\gcd(\text{nrd}(I), \text{nrd}(H_i)) = 1$ (these are coprime, since $N(I) = g$ was chosen as a divisor of T, which is coprime to the norm of ω).

4.5 Computing the Class Group Action

When given an element of $\text{Cl}(\mathfrak{O})$, whose action we wish to evaluate, we first find a smooth representative, as explained in Sect. 4.3. Once such a representing ideal

$$\mathfrak{a} = \prod_{i=1}^{N} \mathfrak{l}_i^{e_i}$$

has been obtained, we need to compute its action. As usual, this is done by repeatedly applying Algorithm 2 on ideals $\mathfrak{a}_0 \mid \mathfrak{a}$ of the form

$$\mathfrak{a}_0 = \prod_{i=1}^{N} \mathfrak{l}_i,$$

until all of $\mathfrak{a}$ has been evaluated. Since the norm of ω is a power of 2 and hence coprime to the primes used in the factor base, this allows Algorithm 2 to be particularly simple, compared to the equivalent algorithms in the original version of SCALLOP [39, Algorithm 2], or SCALLOP-HD [24, Algorithm 3].

Algorithm 2. GroupAction$(\mathfrak{a}, E, P, Q)$

Input: A smooth $\mathfrak{O}$-ideal $\mathfrak{a} = \prod_{i=1}^{N} \mathfrak{l}_i$, an elliptic curve E oriented by $\mathfrak{O}$, and points $P, Q \in E$ generating ϕ_P, ϕ_Q such that $\widehat{\phi_Q} \circ \phi_P$ is an endomorphism corresponding to an element of $\mathfrak{O}$ of norm 2^e

Output: An effectively oriented curve $(E_\mathfrak{a}, P_\mathfrak{a}, Q_\mathfrak{a})$

Let B_1, B_2 be a basis of $E[L]$, where $L = \prod \ell_i$ with $\ell_i = N(\mathfrak{l}_i)$
Let $\widehat{\omega} := \widehat{\phi_P} \circ \phi_Q$
Compute $B_1' = \widehat{\omega}(B_1), B_2' = \widehat{\omega}(B_2)$.
for $i \in \{1, \ldots, N\}$ **do**
  Compute $K_i := [L/\ell_i]([\lambda_i]B_1 + B_1')$, where $\mathfrak{l}_i = (\lambda_i + \omega, \ell_i)$.
  if $K_i = \infty$ **then**
    Compute $K_i := [L/\ell_i]([\lambda_i]B_2 + B_2')$.
  end if
end for
Compute $\phi_\mathfrak{a} : E \to E_\mathfrak{a}$ from its kernel $K = \langle K_1, K_2, \ldots, K_N \rangle$.
return $E_\mathfrak{a}, \phi_\mathfrak{a}(P), \phi_\mathfrak{a}(Q)$

As an optimization, we also present an even simpler, but probabilistic group action evaluation algorithm below. This is based on a standard CSIDH-optimization that avoids sampling points of full order. Since Algorithm 2 requires computing full torsion bases, this optimized sampling is particularily suitable for in SCALLOP.

Algorithm 3. GroupActionOptimized$(\mathfrak{a}, E, P, Q)$

Input: A smooth $\mathfrak{O}$-ideal $\mathfrak{a} = \prod \mathfrak{l}_i$, an elliptic curve E oriented by $\mathfrak{O}$, and points $P, Q \in E$ generating ϕ_P, ϕ_Q such that $\widehat{\phi_Q} \circ \phi_P$ is an endomorphism corresponding to an element of $\mathfrak{O}$ of norm 2^e

Output: An effectively oriented curve $(E'_{\mathfrak{a}}, P_{\mathfrak{a}'}, Q_{\mathfrak{a}'})$, and $N(\mathfrak{a}')$, where $\mathfrak{a}' \mid \mathfrak{a}$

1: Compute $K := [\frac{p+1}{L}]K_0$, where K_0 is a random point on E and $L = \prod N(\mathfrak{l}_i)$
2: Let $\widehat{\omega} := \widehat{\phi_P} \circ \phi_Q$
3: Compute $K' := \widehat{\omega}(K)$
4: Compute $K_{\mathfrak{a}'} := [\lambda]K + K'$, where $\mathfrak{a} = (\lambda + \omega, L)$.
5: Set $L' := \mathrm{ord}(K_{\mathfrak{a}'})$
6: Compute $\phi_{\mathfrak{a}'} : E \to E_{\mathfrak{a}'}$ from its kernel $\langle K_{\mathfrak{a}'} \rangle$
7: **return** $E_{\mathfrak{a}'}, \phi_{\mathfrak{a}'}(P), \phi_{\mathfrak{a}'}(Q), L'$.

4.6 Implementation

We implemented a proof-of-concept version of PEARL-SCALLOP with our parameters in C++. The parameters used can be found in the repository https://www.github.com/biasse/SCALLOP-params. Our implementation applies some well-known, standard optimizations. We work with Montgomery curves

$$E : y^2 = x^3 + Ax^2 + x,$$

and for computing the 2^e-isogeny corresponding to the orientation, we use the formula for 4-isogenies together with optimal strategies following the SIKE documentation [42]. For evaluating the group action, we use Algorithm 3.

We give timings for evaluating a group element, and compare with the timings reported in SCALLOP and SCALLOP-HD in Table 1. The timings for PEARL-SCALLOP were measured an Intel Core i5-1038NG7 CPU, clocked at 2.00GHz The timings are given as the average time of evaluating 10 random group elements. As concluded in Sect. 3, computing parameters for security levels CSIDH-2048 and 4096 is feasible, but costly. Hence, we choose to only provide timings for 512, 1024 and 1536, as these already show a significant improvement over SCALLOP and SCALLOP-HD.

Note that all three implementations are proof-of-concept with non-optimized code, so the timings provide only an approximate comparison. In particular, the timings for SCALLOP-HD are based on a SageMath [56] implementation, rather than C++. Although SCALLOP-HD's GitHub repository contains data

Table 1. Timings from SCALLOP [39, Section 6.2], SCALLOP-HD [24, Section 5.6] and PEARL-SCALLOP.

Security level	SCALLOP	SCALLOP-HD	PEARL-SCALLOP
CSIDH-512	35 s	1 min, 28 s	30 s
CSIDH-1024	12 m, 30 s	19 min	58 s
CSIDH-1536	–	–	11 min, 50 s

for all security levels, timings were only reported for the two lowest levels in [24]. However, a theoretical efficiency comparison of PEARL-SCALLOP and SCALLOP-HD is in fact easy. The prime p of the base field can be chosen almost identical, and the group action evaluation is very similar, except in SCALLOP-HD it requires the evaluation of a $(2^e, 2^e)$-isogeny between abelian surfaces, while PEARL-SCALLOP relies on the evaluation of a 2^e-isogeny between elliptic curves.

5 Conclusion

In this work, we presented PEARL-SCALLOP, a new way of instantiating an efficient cryptographic group action based on SCALLOP [39]. In contrast to SCALLOP, our technique is feasible to instantiate for higher security levels, employs a significantly more efficient group action evaluation, and is based on a different hardness assumption.

SCALLOP-HD [24], another efficient cryptographic group action based on SCALLOP, does allow instantiations at higher security levels. However, in comparison, PEARL-SCALLOP is again more efficient in terms of group action evaluation and is based on a different hardness assumption. It also permits practical instantiations for security levels equivalent to CSIDH-4096. We hence argue that PEARL-SCALLOP is currently the ideal choice for efficiency, while also allowing realistic instantiations for secure parameter levels.

Acknowledgements. This project started at the 2023 Banff workshop on isogeny-based cryptography, and we want to thank the organisers for a successful workshop. We also want to thank Gioella Lorenzon and Frederik Vercauteren for useful comments and feedback on an earlier version of this paper.

B. Allombert and A. Page were supported by ANR CIAO ANR-19-CE48-0008 and PEPR Cryptanalyse ANR-22-PECY-0010. R. Scheidler was supported by NSERC of Canada. This research was supported by the Ministry of Culture and Innovation and the National Research, Development, and Innovation Office within the Quantum Information National Laboratory of Hungary (Grant No. 2022-2.1.1-NL-2022-00004) and by the grant "EXCELLENCE-151343". Péter Kutas is supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the UNKP-23-5 New National Excellence Program. Péter Kutas is also partly supported by EPSRC through grant number EP/V011324/1. J. K. Eriksen was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT - No. 101020788), by the Research

Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VR20192203.

References

1. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12492, pp. 411–439. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_14
2. Albrecht, M.R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E.W., Stevens, M.: The general sieve kernel and new records in lattice reduction. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 717–746. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_25
3. Albrecht, M.R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E.W., Stevens, M.: The General Sieve Kernel (G6K) (2019)
4. Arpin, S., Clements, J., Dartois, P., Eriksen, J.K., Kutas, P., Wesolowski, B.: Finding orientations of supersingular elliptic curves and quaternion orders. Cryptology ePrint Archive, Paper 2023/1268 (2023). https://eprint.iacr.org/2023/1268
5. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
6. Bach, E.: Explicit bounds for primality testing and related problems. Math. Comp. **55**(191), 355–380 (1990)
7. Bernstein, D.J., De Feo, L., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. In: ANTS XIV—Proceedings of the Fourteenth Algorithmic Number Theory Symposium, vol. 4 of Open Book Series, pp. 39–55. Math. Sci. Publ., Berkeley (2020)
8. Bernstein, D.J., Lange, T., Martindale, C., Panny, L.: Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 409–441. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_15
9. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: generic, simple, and efficient. In: EUROCRYPT 2022. Part II, vol. 13276, pp. 95–126. Springer, Cham (2022)
10. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12492, pp. 464–492. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_16
11. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: efficient isogeny based signatures through class group computations. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11921, pp. 227–247. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34578-5_9
12. Biasse, J.-F.: Improvements in the computation of ideal class groups of imaginary quadratic number fields. Adv. Math. Commun. **4**(2), 141–154 (2010)
13. Biasse, J.-F., Fieker, C.: Subexponential class group and unit group computation in large degree number fields. LMS J. Comput. Math. **17**, 385–403 (2014)
14. Biasse, J.-F., Fieker, C., Jacobson, M.J., Jr.: Fast heuristic algorithms for computing relations in the class group of a quadratic order, with applications to isogeny evaluation. LMS J. Comput. Math. **19**, 371–390 (2016)

15. Bonnetain, X., Schrottenloher, A.: Quantum security analysis of CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12106, pp. 493–522. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_17
16. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symb. Comput. **24**(3-4), 235–265 (1997). https://www.math.ru.nl/~bosma/pubs/JSC1997Magma.pdf
17. Bottinelli, P., de Quehen, V., Leonardi, C., Mosunov, A., Pawlega, F., Sheth, M.: The dark SIDH of isogenies. Cryptology ePrint Archive, Paper 2019/1333 (2019). https://eprint.iacr.org/2019/1333
18. Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., Zimmermann, P.: Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12171, pp. 62–91. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_3
19. Castryck, W., Houben, M., Vercauteren, F., Wesolowski, B.: On the decisional Diffie-Hellman problem for class group actions on oriented elliptic curves. Res. Number Theory **8**(4), Paper No. 99, 18 (2022)
20. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018. LNCS, vol. 11274, pp. 395–427. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_15
21. Castryck, W., Meeren, N.V.: Two remarks on the vectorization problem. Cryptology ePrint Archive, Paper 2022/1366 (2022). https://eprint.iacr.org/2022/1366
22. Castryck, W., Sotáková, J., Vercauteren, F.: Breaking the decisional diffie-hellman problem for class group actions using genus theory. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12171, pp. 92–120. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_4
23. Chen, M., Imran, M., Ivanyos, G., Kutas, P., Leroux, A., Petit, C.: Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of psidh. Cryptology ePrint Archive, Paper 2023/779 (2023). https://eprint.iacr.org/2023/779
24. Chen, M., Leroux, A., Panny, L.: SCALLOP-HD: group action from 2-dimensional isogenies. Cryptology ePrint Archive, Paper 2023/1488 (2023). https://eprint.iacr.org/2023/1488
25. Childs, A.M., van Dam, W.: Quantum algorithms for algebraic problems. Rev. Mod. Phys. **82**(1), 1 (2010)
26. Colò, L., Kohel, D.: Orienting supersingular isogeny graphs. J. Math. Cryptol. **14**(1), 414–437 (2020)
27. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Paper 2006/291 (2006). https://eprint.iacr.org/2006/291
28. Cox, D.A.: Primes of the form x^2+ny^2—Fermat, Class Field Theory, and Complex Multiplication, 3rd edn. AMS Chelsea Publishing, Providence (2022)
29. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQISignHD: new dimensions in cryptography. Cryptology ePrint Archive, Paper 2023/436 (2023). https://eprint.iacr.org/2023/436
30. De Feo, L., Galbraith, S.D.: SeaSign: compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 759–789. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_26
31. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the Deuring correspondence: towards practical and secure SQISign signatures. In: EUROCRYPT 2023. Part V, vol. 14008, pp. 659–690. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_23

32. De Feo, L., Meyer, M.: Threshold schemes from isogeny assumptions. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020. LNCS, vol. 12111, pp. 187–212. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45388-6_7
33. de Quehen, V., Kutas, P., Leonardi, C., Martindale, C., Panny, L., Petit, C., Stange, K.E.: Improved torsion-point attacks on SIDH variants. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12827, pp. 432–470. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84252-9_15
34. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
35. Doulgerakis, E., Laarhoven, T., de Weger, B.: Finding closest lattice vectors using approximate voronoi cells. In: Ding, J., Steinwandt, R. (eds.) PQCrypto 2019. LNCS, vol. 11505, pp. 3–22. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_1
36. Eisenträger, K., Hallgren, S., Lauter, K., Morrison, T., Petit, C.: Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 329–368. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_11
37. Eriksen, J.K., Leroux, A.: Computing orientations from the endomorphism ring of supersingular curves and applications. Cryptology ePrint Archive, Paper 2024/146 (2024). https://eprint.iacr.org/2024/146
38. Eriksen, J.K., Panny, L., Sotáková, J., Veroni, M.: Deuring for the people: supersingular elliptic curves with prescribed endomorphism ring in general characteristic. Cryptology ePrint Archive, Paper 2023/106 (2023). https://eprint.iacr.org/2023/106
39. Feo, L.D., Fouotsa, T.B., Kutas, P., Leroux, A., Merz, S.P., Panny, L., Wesolowski, B.: SCALLOP: scaling the CSI-FiSh. In: Public-key cryptography—PKC 2023. Part I, vol. 13940, pp. 345–375. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-31368-4_13
40. Hafner, J.L., McCurley, K.S.: A rigorous subexponential algorithm for computation of class groups. J. Am. Math. Soc. **2**(4), 837–850 (1989)
41. Ivanyos, G.: On solving systems of random linear disequations (2007). https://arxiv.org/abs/2401.16644
42. Jao, D., et al.: SIKE. Technical report, National Institute of Standards and Technology (2022). https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
43. Joux, A., Odlyzko, A., Pierrot, C.: The past, evolving present, and future of the discrete logarithm. In: Koç, Ç.K. (ed.) Open Problems in Mathematics and Computational Science, pp. 5–36. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10683-0_2
44. Klüners, J., Pauli, S.: Computing residue class rings and Picard groups of orders. J. Algebra **292**(1), 47–64 (2005)
45. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM J. Comput. **35**(1), 170–188 (2005)
46. Leroux, A.: A new isogeny representation and applications to cryptography. In: International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2022. Part II, vol. 13792, pp. 3–35. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-22966-4_1
47. Leroux, A.: Quaternion Algebra and isogeny-based cryptography. PhD thesis, PhD thesis, Ecole doctorale de l'Institut Polytechnique de Paris (2022)
48. Onuki, H.: On oriented supersingular elliptic curves. Finite Fields Appl. **69**, Paper No. 101777, 18 (2021)

49. Peikert, C.: He gives C-sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12106, pp. 463–492. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_16
50. Schnorr, C.P., Euchner, M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Program. **66**(2), 181–199 1994
51. Petit, C.: Faster algorithms for isogeny problems using torsion point images. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10625, pp. 330–353. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_12
52. Rostovtsev, A., Stolbunov, A.: Public-keycryptosystems based on isogenies. Cryptology ePrint Archive, Paper 2006/145 (2006). https://eprint.iacr.org/2006/145
53. Shoup, V., et al.: NTL: a library for doing number theory (2001). https://libntl.org/
54. The CADO-NFS Development Team. CADO-NFS, an implementation of the number field sieve algorithm. Release 2.3.0 (2017)
55. The PARI Group, Univ. Bordeaux. PARI/GP version texttt2.16.1 (2022). http://pari.math.u-bordeaux.fr/
56. The Sage Developers. SageMath, the Sage Mathematics Software System (version 9.7) (2022). https://sagemath.org
57. Wesolowski, B.: Orientations and the supersingular endomorphism ring problem. In: Advances in cryptology—EUROCRYPT 2022. Part III, vol. 13277, pp. 345–371. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07082-2_13